Datorteknik DT509G Home Exam

August 19, 2020

- This time the exam is slightly different: you have a take-home exam. This means that you are allowed to use the book and search for reference material on the internet.
- If you choose to use online materials, you **need to reference them**. Not referencing a source will result in an immediate fail of the whole exam.
- You are not allowed to use group discussions to solve the exercises. Suspected cases of copying from another student will result in an immediate fail of the whole exam and referal to the academic integrity board.
- This exam has been personally created for you, so provide answers only to the questions you receive here.
- You can use any typesetting software (word, open office, google docs, latex) or write by hand and scan the paper. The final submission should be in the form of 1 single PDF file.
- You have 4 hours to solve the exam. Late submissions will not be accepted.
- The exam is 50 points, 25 points are needed to pass.
- If you have questions, write a blackboard message to Todor.

1 Logic, Representations and Architectures (5 points)

a: When a computer architect refers to a modified Harvard architecture, what do they mean? How does that differ from the classical Von Neumann and Harvard architectures and what are the advantages? (3 points)

b: What is a latch circuit and what is it useful for? How can we implement a latch using boolean gates? (2 points)

2 Processors (15 points)

For the next exercises, assume you are programming in assembly on a 16-bit architecture which employs a five-stage pipeline for instruction execution. The five stages are:

(1) fetch next instruction

(2) decode instruction and fetch operands

(3) perform ALU operation

(4) read or write to memory

(5) store result in register

Your architecture has 8 general purpose registers split into two banks with $r0 - r3$ on bank 1 and $r4 - r7$ on bank 2. The following instructions are implemented:

load loads an immediate value into a register r_a (r_a and the immediate values are provided as arguments)

mov moves a value from a register r_a to a register r_b

add adds the value in register r_a to the value in register r_b and stores the result in register r_b

sub subtracts the value in register r_b from the value in register r_a and stores the result in register r_b

cmp compares the value of registers r_a and r_b and sets the global register *cmp* to 1 is $r_a \geq r_b$ and to 0 otherwise

bne increments the program counter *pc* by an immediate value if the value of the global register *cmp* is not equal to 1.

jmp sets the program counter *pc* to a value $r_a + offset$ and moves on to the next instruction.

a: Assuming that each stage of the pipeline takes three clock cycles to complete, how many instructions per clock cycle does the overall architecture execute? Motivate your answer and explain under what conditions it holds true. What would be the consequence of improving stages (1), (2) and (3) to process one instruction per cycle? (3 points)

b: For which instructions in the ISA does the use of register banks speed up execution? (2 points)

You have written the following assembly code

```
load   r0,10        # load value 10 into register r0
load   r1,1         # load immediate value 1 into register r1

load   r2,0         # load zero into register r2
load   r3,0         # load 0 into register r3
load   r4,1         # load 1 into register r4
load   r5,0         # load 0 into r5
while:
add    r1,r2        # add contents of r1 and r2, store into r2
mov    r4,r5        # r5=r4
add    r3,r4        # r4 = r3+r4
mov    r5,r3        # r3=r5
cmp    r2,r0        # cmp = r2>=r0
bne  while
```

c: How many pipeline stalls will occur while executing this program? (2 points)

d: Design a binary encoding for the instruction set. What type of operand encoding are you using? What is the largest offset that you can encode? (3 points)

e: Explain what a two-pass assembler is and illustrate how it would produce a binary code from the above assembly code. (3 points)

f: Assume you want to print out the value of r5 at the end of the loop by interfacing to the C standard library. What would you need to do in order to connect your assembly code and how would you refernce the function?

3 Memory (10 points)

a: What are the requirements for a component to qualify as a memory cache? (2 points)

b: Direct Mapped Memory Cache is implemented on a 16-bit architecture. The chache can hold 8 blocks of memory, each of which is 32 bytes long. How many bits are

necessary in order to represent the tag of each of these blocks? Explain how the cache works and how powers of two are used to parse a byte address and find the correct data stored in the cache. (3 points)

c: What is paging? Give three examples of paging systems. How does demand paging perform address translation? (3 points)

d: What is memory alignment and when is it relevant? How does a non-aligned variable likely to deteriorate cache performance? (2 points)

4 I/O architecture (10 points)

a: A printer's control and status register (CSR) is represented by a single byte with the following control and status bits:

- b_0 true if printer is idle,
- b_1 true if paper has been moved one line forward,
- b_2 true if ready to print next line,
- b_3 if set to true makes printer move paper one line forward,
- b_4 if true printer writes next line from data register onto page.

The remaining bits are not used. Write pseudocode for printing a page of text using polling. (3 points)

b: The exercise above uses polling for controlling a printer. What are the disadvantages of using polling? What are the alternatives? How would the code from the previous example need to change if you want to use a more efficient way to control the I/O device? How would the hardware need to change? (4 points)

c: Explain what are the main advantages of Direct Memory Access (DMA). How can buffer and operation chaining be used to improve DMA performance? (3 points)

5 Open Question (10 points)

Imagine that you are writing a mini operating system for an Arduino Uno board (`https://store.arduino.cc/arduino-uno-rev3`). You got to the point where you need to interface with a keyboard that runs over a serial connection and want to define a device driver. Look up how you would be able to set up an interrupt for reading events coming from the keyboard. Explain what would be the necessary parts to implement and which functions you would use to read information from the device.